Stochastic evolution and multifractal classification of prokaryotes

Matthew J. Berryman[a], Andrew Allison[a], and Derek Abbott[a]

[a]Center for Biomedical Engineering and
School of Electrical and Electronic Engineering,
The University of Adelaide, SA 5005, Australia

ABSTRACT

We introduce a model for simulating mutation of prokaryote DNA sequences. Using that model we can then evaluated traditional techniques like parsimony and maximum likelihood methods for computing phylogenetic relationships. We also use the model to mimic large scale genomic changes, and use this to evaluate multifractal and related information theory techniques which take into account these large changes in determining phylogenetic relationships.

Keywords: phylogenetic trees, DNA sequences, mutations, evolution, multifractal

1. INTRODUCTION

In this paper we examine the stochastic evolution of prokaryotes by simulating a number of mutational events from changes within genes to changes in overall genome structure. We then use the resultant *in silico* virtual mutants,[1] for which we know the ancestry, to compare the accuracy of both whole genome comparisons and more traditional orthologous gene comparisons in deriving phylogenetic relationships.

The relationships between various prokaryotes, bacteria and archaea, are of great interest.[2, 3] Since the seminal work by Zuckerkandl and Pauling,[4] many of these relationships have been explored by comparing the DNA and amino acid sequences of the organisms in question. Given a set of sequences from several species, we would then like to infer (with statistical significance) a phylogenetic relationship between the species. A critical assumption is that the sequences diverged from a common ancestor and not by other processes such as gene duplication[5] or gene transfer,[6] these are known as orthologues. The algorithm used to analyze the sequence needs to produce a measure of divergence from the common ancestor which can then be used to construct a variety of trees.[7]

A number of processes are well known by which prokaryotes can mutate and thus diverge from a common ancestor.[8] The main mechanisms we have focused on are:

- Base substitutions, where one base pair has been replaced with a different base through some mechanism (such as UV irradiation with an absent or unsuccessful repair process).
- Additions and deletions, where a base pair has been added or removed from the sequence.
- Rearrangements, where a sequence has been inverted or shifted to another location (or both).
- Gene transfer, where one or more genes are inserted into a prokaryote genome from another source (such as a phage or by recombination with a plasmid).

Send correspondence to Derek Abbott
E-mail: dabbott@eleceng.adelaide.edu.au, Telephone: +61 8 8303 5748

In addition to analyzing orthologous genes, one can compare whole genomes and their features and properties due to the large number of complete prokaryote genomes available.[9] There have been a number of methods proposed for using a whole genome approach to problems of phylogeny. These include measuring the fraction of orthologs shared between genomes and quantifying correlations between genes with respect to their relative positions in genomes.[10] Similarly, comparing orderings of genes between genomes has been proposed as an area of investigation.[11] Other techniques which we have explored in this paper include a multifractal approach[12] and related information theory approaches.[13, 14]

2. RESULTS

2.1. Comparison of standard algorithms

We have compared a known tree that we have generated by simulating mutation of the human adenosine a2 receptor[15] with trees produced by standard techniques such as maximum likelihood[7, 16] and maximum parsimony.[7]

The basic algorithm of our mutation simulation software is as follows:

1. Take the original nucleotide sequence, and apply up to five insertions, up to five deletions, and up to five base substitutions. We use a small random number of each to simulate random mutations and to put some distance between each of the generations, but not too much that it becomes too easy for the algorithms to distinguish between them. Repeat this step n times to produce n descendants of the original, these we called $0, \ldots, n-1$.
2. Taking the mutations of the previous step, x, we then mutate these to produce some m descendants of those descendants, which we label $x \oplus 0, \ldots, x \oplus (m-1) \forall x$ (where $\oplus$ denotes concatenation).
3. Repeat the previous step several times.

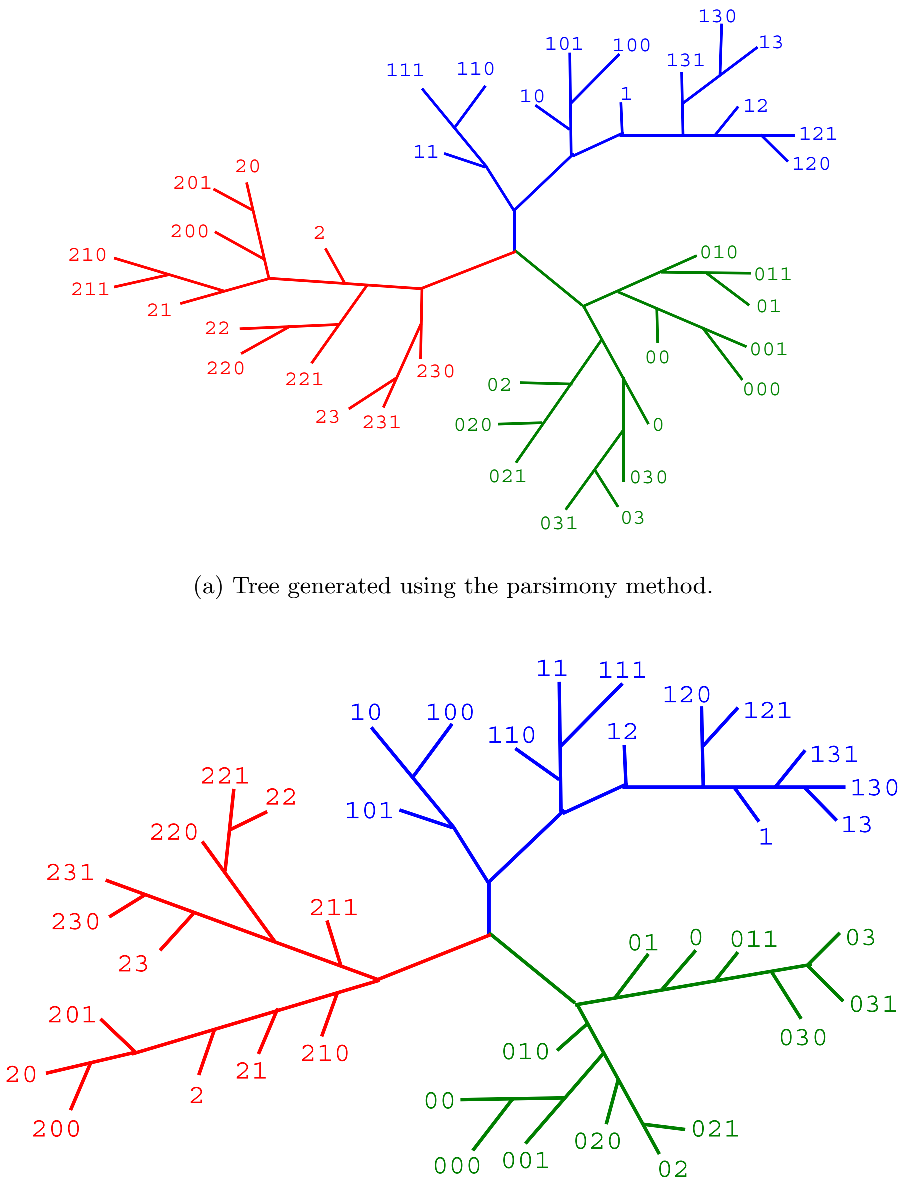
We then use the CLUSTALW software[17, 18] to perform the alignments. The algorithm, as described in Durbin *et al.*[7] is:

1. Construct a distance matrix of all pairs of sequences by a pairwise dynamic programming alignment, followed by conversion of similarity scores to approximate evolutionary distances using the Kimura model.[19]
2. Construct a guide tree by a neighbor-joining clustering algorithm of Saitou and Nei.[20]
3. Progressively align at nodes in order of decreasing similarity, using sequence-sequence, sequence-profile, and profile-profile alignment.

After alignment, we then generated the trees shown in Figure 1. We used the PHYLIP software[21] which implements the two following methods:

- Parsimony, which generates the tree by evaluating a number of possible trees and finding the one with the overall minimum cost, where the cost is the number of substitutions to explain the observed sequences.
- Maximum likelihood, which builds a tree with a maximum likelihood of occurring given a model of evolution and the observed sequences. The critical assumptions of the model are that base substitutions (and gaps) follow a Poisson process with a set of specified rates, that each site in the sequence evolves independently, and different lineages evolve independently.

Both techniques worked well, however we felt that the parsimony technique worked better than the maximum likelihood technique. A number of open questions have been raised about the efficacy of gap penalty scoring.[22, 23] In order to see if the incorrect portions of the tree were due to incorrect scoring of gaps, we repeated the steps outlined above but with no gap-producing insertions and deletions and with a corresponding increase in the number of substitutions (to make the distances comparative). We found no significant difference in the trees produced, suggesting the problems lie elsewhere.



(a) Tree generated using the parsimony method.

(b) Tree generated using the maximum likelihood method. The tree is constructed to give a maximum likelihood score, and for this tree the log likelihood score is -4074.2. So given a probabilistic model, the probability that tree fits that model is $e^{-4074.2}$, which although low is higher than trees with minor variations that have log likelihood scores in the range -4075 to -4080.

Figure 1. This shows two trees generated using the parsimony and maximum likelihood methods. A correct tree should have the numbers increasing in size going outwards from the center, with each descendant of x being $x \oplus n$, for some n.

2.2. Multifractal classification

Here we use the fractal method as detailed by Yu *et al.*,[12, 24] which considers the Rényi dimension D_q for $q \in \mathbf{R}$, given by

$$D_q = \begin{cases} \lim\limits_{\epsilon \to 0} \dfrac{\ln Z_\epsilon(q)}{(q-1)\ln \epsilon}, & q \neq 1, \\ \\ \lim\limits_{\epsilon \to 0} \dfrac{Z_\epsilon(q)}{\ln \epsilon}, & q = 1 \end{cases} \tag{1}$$

where

$$Z_\epsilon(q) = \begin{cases} \sum\limits_{\mu(B) \neq 0} [\mu(B)]^q, & q \neq 1, \\ \\ \sum\limits_{\mu(B) \neq 0} \mu(B) \ln \mu(B), & \text{otherwise.} \end{cases} \tag{2}$$

The $\mu(B)$ are simply the sample probabilities (in $[0, 1]$) of finding each of the possible substrings of codons of length K, ϵ is $\epsilon = 4^{-K}$. We found a value of $K = 8$ works best. The space (D_{-1}, D_1, D_{-2}) used in conjunction with the neighbor joining algorithm[20] is then used to generate phylogenetic trees.

Since the multifractal technique deals with large scale statistics, it is unreliable in dealing with the short sequences we analysed in the previous subsection. The question then arises, is the technique of any use as a whole-genome classification technique? To analyze this question, we model larger changes to bacterial genomes, in particular both gene transfer[6] and shuffling of operons to different positions on the genome.[11] We have focussed on the *Escherichia coli* bacteria, due to the considerable amount of detailed information available on their genetic makeup[25] and also the mechanisms by which genetic material can be inserted into their genomes.[8] The *E. coli* bacteria we used were K12 (Genbank accession number NC_000913), O157:H7 (NC_002695), O157:H7 EDL933 (NC_002655), and CFT073 (NC_004431). In the rest of this paper we simply use the numerical part of the accession numbers to refer to the different subspecies. The *E. coli* bacteria have a large, variable portion of foreign genes[6] so they provided an excellent test for a technique which is good at picking up differences in gene usage.

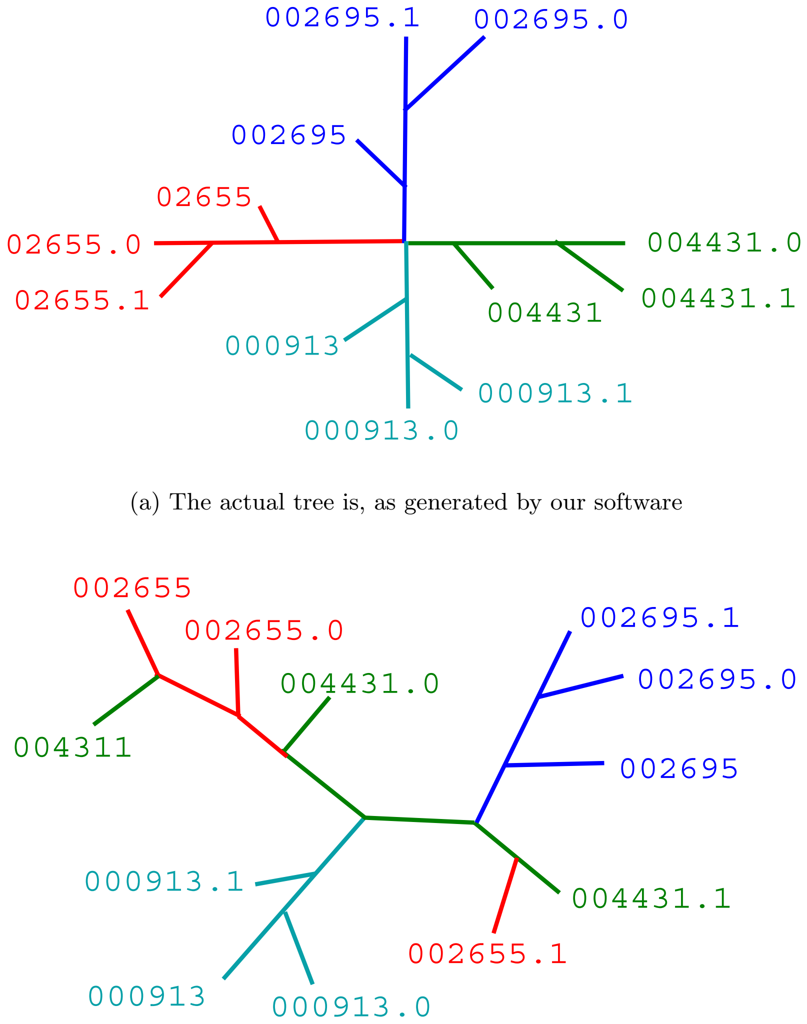
The particular algorithms we used for moving entire genes around are:

- Gene shuffling. Here we identified operons based on existing information about operons in *E. coli* bacteria which details predicted operons in *E. coli.*[26] Since we wanted to see how the multifractal method copes with shifts in general (it does not in fact distinguish between the different types of shifting that occur), it is unimportant if there are some errors in finding operons. For future work on whole genome analysis, especially that considering the ordering of genes,[11] it may be necessary to significantly improve this rate if our simulator is to accurately represent actual biological processes.

- Recombination. Here we identified regions where *E. coli* restriction enzymes could cleave the sequence, and then picked them at random to insert one or more sequences. Both the enzymes used and the additional sequences could be specified by the user (for example, one could insert the adenosine a2 receptor gene used in the previous subsection and insert that using the Eco57I restriction enzyme[27]).

As before, we also introduced a number of base substitutions, insertions, and deletions into the sequence, this time we used a random number of each of up to thirty.

We then considered both the parsimony and maximum likelihood methods for comparing a selection of both DNA and amino acid sequences from a set of derived mutants, and used the multifractal with neighbor joining method to compare the whole genome sequences. The results of using the multifractal method this are shown in Figure 2.



(a) The actual tree is, as generated by our software

(b) This is the tree as computed using the multifractal measure and the neighbor joining algorithm. Note that the tree is correct for the 002695 and 000913 families, but has significant problems distinguishing between the 002655 and 004431 families.

Figure 2. This shows two trees generated using the parsimony and maximum likelihood methods. A correct tree should have the numbers increasing in size going outwards from the center, with descendants of x labelled $x.n$, $n = 0, 1$.

2.3. Evalution of multifractal methods

As can be seen in Figure 2, our earlier work,[28] and that of Yu *et al.*,[12] the multifractal technique has some promise in classifying bacteria and generating phylogenetic trees but has difficulties distinguishing between closely related bacteria. This is due to the fact that it ignores information differences in structure between different genomes and also the features of the genomes such as genes and repeat sequences. To determine an accuracy rate, we simulated large scale genomic changes as in the previous section for both the *E. coli* bacteria previously used as well as for a set of quite different bacteria which consisted of *Bacillus cereus*, *Shigella flexneri*, *Salmonella typhimurium*, *Pseudomonas aeruginosa*, *Mycoplasma pulmonis*, *Lactobacillus plantarum*, and *Bradyrhizobium japonicum.* In addition to simply benchmarking against trees produced by our software, we also considered well established relationships between bacteria.[29, 30] For each tree we determined an accuracy rate by considering the fraction of the tree that was correct and the number of basic tree operations (rotates and shifts) needed to correct the tree. The results are given in Table 1 and gave an average of 56% correct.

Table 1. The fraction of the organisms with a correct relationship to each other in the trees are shown for a number of simulated sets of evolution and a known set of bacteria. The number of tree operations (shifts and rotates) required to turn the output tree into the correct tree are shown. Simulation run **8** is for the set of actual bacteria with a known relationship, rather than a simulated set of bacteria.

Simulation run	**% of tree in correct positions**	**Tree op's required to correct**
1	50%	4
2	33%	7
3	58%	5
4	83%	2
5	50%	7
6	42%	5
7	75%	3
8*	57%	4

Overall the multifractal technique is useful in determining rough relationships between organisms, but has trouble coping with finer details, most likely because it doesn't use enough of the information available about the genomes in question. In the following subsection we detail a related information theory approach.

2.4. Asymptotically optimal universal compression metrics

Here we use the same set of files we generated in subsection 2.2 and apply a lossless compression algorithm, with a coding rate that approaches the Shannon rate[31, 32]:

$$H\left(P\right) = -\sum_{p_i \in P} p_i \log p_i \tag{3}$$

The Shannon entropy is in fact a special case of the more general Rényi dimension 1 used in Subsection 2.2.

We use the metric

$$M = H(X|Y)^2 = \left(H(X) - H(X,Y)\right)^2, \tag{4}$$

where

$$H(X,Y) = \lim_{(n,N)\to\infty} \frac{E\left[L\left(x_1^n \oplus y_1^N\right)\right]}{n+N} \tag{5}$$

and

$$H(X) = \lim_{n\to\infty} \frac{E\left[L\left(x_1^n\right)\right]}{n}. \tag{6}$$

In Equations 5 and 6, x_1^n and y_1^N are the sequence strings of the two genomes of lengths n and N bits (two bits per base), and $E\left[L\left(\cdot\right)\right]$ is the compressed length operator (again, in units of bits). The tree for the same mutations as in Subsection 2.2 is shown in Figure 3. As with the multifractal measure, the information theory measure has difficulties determining differences between closely related bacteria.

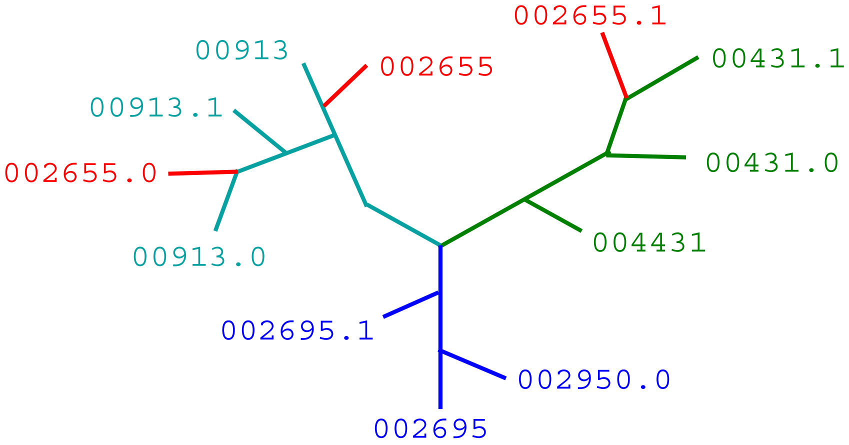


Figure 3. This is the tree produced by using an information theory based metric in conjunction with the neighbor joining algorithm. As with the multifractal tree, it has trouble distinguishing between closely related genomes, for similar reasons. We propose tightening the tree building decision methods using techniques outlined in Gutman.[13] This will enable us to place accuracy levels on parts of the tree.

3. CONCLUSIONS

In general we found that the multifractal and compression metrics we have explored are useful in distinguishing between different organisms, they have problems distinguishing between those which are closely related, or those which are too small for statistical properties to be estimated with any accuracy. We found the popular maximum likelihood and parsimony techniques to handle small changes in short sequences quite well, with the parsimony tree building method handling gaps better than maximum likelihood. We used an entirely automated approach to the problem of multiple alignment, using the CLUSTALW software package. A skilled operator would be able to edit the CLUSTALW results to produce better alignments which would help the maximum likelihood and parsimony algorithms further.

Although the multifractal and compression metrics were found to be useful, the fact that they ignore a considerable amount of whole-genome information available such as gene reversals and gene ordering between different species means they will always have certain limitations in accuracy. Work can be done to specify the accuracy of the metrics produced by these methods. In particular, the compression method can identify organisms that it is unable to classify reliably and this could be indicated when drawing trees. the way ahead in whole genome comparisons and the resultant phylogenies lies with combining multiple techniques including, but not limited to, comparison of sets of aligned protein sequences, comparison of sets of *unaligned* protein sequences,[33] and information on gene order changes.[11,34] Some of these techniques can be time consuming, however, requiring significant amounts of human input. Perhaps the best approach would be to combine whole genome techniques with those of orthologous gene comparisons and even physiological data in determining highly accurate phylogenies.

ACKNOWLEDGMENTS

We gratefully acknowledge funding from The University of Adelaide.

REFERENCES

1. J. Kwasigroch, D. Gilis, Y. Dehouck, and M. Rooman, “PoPMuSiC, rationally designing point mutations in protein structures,” *Bioinformatics* **18**, pp. 1701–1702, 2002.
2. M. Pagel, “Inferring the historical patterns of biological evolution,” *Nature* **401**, pp. 877–884, Oct. 1999.

3. M. Woolhouse, J. Webster, E. Domingo, B. Charlesworth, and B. Levin, "Biological and biomedical implications of the co-evolution of pathogens and their hosts," *Nature Genetics* **32**, pp. 569–577, Dec. 2002.
4. E. Zuckerkandl and L. Pauling, *Molecular disease, evolution and genetic heterogeneity*, pp. 189–225. Academic Press, 1962.
5. S. Ohno, *Evolution by Gene Duplication*, Springer-Verlag, 1970.
6. H. Ochman, J. Lawrence, and E. Groisman, "Lateral gene transfer and the nature of bacterial innovation," *Nature* **405**, pp. 299–304, 2000.
7. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1998.
8. F. Joset and J. Guespin-Michel, *Prokaryotic Genetics: Genome Organisation, Transfer and Plasticity*, Blackwell Scientific Publications, 1993.
9. R. Doolittle, "Microbial genomes multiply," *Nature* **416**, pp. 697–700, Apr. 2002.
10. M. Huynen and P. Bork, "Measuring genome evolution," *Proc. Natl. Acad. Sci. USA* **95**, pp. 5849–5856, May 1998.
11. J. Tamames, "Evolution of gene order conservation in prokaryotes," *Genome Biology* **2**, pp. research0020.1–0020.11, June 2001.
12. V. Anh, K. Lau, and Z. Yu, "Multifractal characterisation of complete genomes," *Journal of Physics A* **34**, pp. 7127–7139, Sept. 2001.
13. M. Gutman, "Asymptotically optimal classification for multiple tests with empirically observed statistics," *IEEE Transactions on Information Theory* **35**, pp. 401–408, Mar. 1989.
14. M. Li, J. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, "An information-based sequence distance and its application to whole mitochondrial genome phylogeny," *Bioinformatics* **17**(2), pp. 149–154, 2001.
15. T. Furlong, K. Pierce, L. Selbie, and J. Shine, "Molecular characterization of a human brain adenosine a2 receptor," *Brain Res. Mol. Brain Res.* **15**, pp. 62–66, 1992.
16. J. Felsenstein, "Evolutionary trees from DNA sequences: A maximum likelihood approach," *Journal of Molecular Evolution* **17**, pp. 368–376, 1981.
17. J. Thompson, D. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice," *Nucleic Acids Research* **22**, pp. 4673–4680, 1994.
18. D. Higgins, J. Thompson, and T. Gibson, "Using CLUSTAL for multiple sequence alignments," *Methods in Enzymology* **266**, pp. 383–402, 1996.
19. M. Kimura, *The Neutral Theory of Molecular Evolution*, Cambridge University Press, 1983.
20. N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Molecular Biology and Evolution* **4**, pp. 406–425, 1987.
21. J. Felsenstein, "Inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods," *Methods in Enzymology* **266**, pp. 418–427, 1996.
22. W. Fitch and T. Smith, "Optimal sequence alignments," *Proc. Natl. Acad. Sci. USA* **80**, pp. 1382–1386, Mar. 1983.
23. J. Reese and W. Pearson *Bioinformatics* **18**(11), pp. 1500–1507, 2002.
24. Z. Yu, V. Anh, and K. Lau, "Measure representation and multifractal analysis of complete genomes," *Physical Review E* **64**, pp. 031903/1–9, Sept. 2001.
25. T. Hayashi, K. Makino, M. Ohnishi, K. Kurokawa, K. Ishii, K. Yokoyama, *et al.*, "Complete genome sequence of enterohemorrhagic Escherichia coli O157:H7 and genomic comparison with a laboratory strain K-12," *DNA Research* **8**, pp. 11–22, 2001.
26. H. Salgado, A. Santos-Zavaleta, S. Gama-Castro, D. Millán-Zárate, F. Blattner, and J. Collado-Vides, "RegulonDB (version 3.0): transcriptional regulation and operon organization in *Escherichia coli* K-12," *Nucleic Acids Research* **28**(1), pp. 65–67, 2000.
27. R. Rimseliene and A. Janulaitis, "Mutational analysis of two putative catalytic motifs of the type IV restriction endonuclease Eco57I," *Journal of Biological Chemistry* **276**, pp. 10492–10497, Mar. 2001.

28. M. Berryman, A. Allison, D. Abbott, and P. Carpena, "Signal processing and statistical methods in analysis of text and DNA," *Proc. SPIE: Biomedical Applications of Micro- and Nanoengineering* **4937**, pp. 231–240, 2002.
29. J. Brown, C. Douady, M. Italia, W. Marshall, and M. Stanhope, "Universal trees based on large combined protein sequence data sets," *Nature Genetics* **28**, pp. 281–285, 2001.
30. G. Olsen, C. Woese, and R. Overbeek, "The winds of (evolutionary) change: breathing new life into microbiology," *Journal of Bacteriology* **176**, pp. 1–6, 1994.
31. A. Lempel and J. Ziv, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory* **23**(3), pp. 337–343, 1977.
32. J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory* **24**(5), pp. 530–536, 1978.
33. G. Stuart, K. Moffett, and S. Baker, "Integrated gene and species phylogenies from unaligned whole genome protein sequences," **18**(1), pp. 100–108, 2002.
34. B. Moret, J. Tang, L.-S. Wang, and T. Warnow, "Steps toward accurate reconstructions of phylogenies from gene-order data," *Journal of Computer and Systems Sciences* **65**(3), pp. 508–525, 2002.